A telerobot to extend the skills of microsurgeons

Hari Das, Tim Ohm, Curtis Boswell, Guillermo Rodriguez, Robert Steele
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA.

Steve Charles
MicroDexterity Systems, Inc. Memphis TN

Abstract - The engineering details of the Robot Assisted MicroSurgery (RAMS) telerobotic system designed to assist microsurgeons improve the precision and dexterity with which they can position surgical instruments is described in this paper. This work is the result of a collaboration between NASA/JPL and MicroDexterity Systems, Inc. Prototypes developed have been used in demonstrations of eye microsurgery and micro-suturing. The force-feedback control scheme used and a shared control pivoting algorithm are described. MicroDexterity Systems, Inc. continues this effort with plans to commercialize the technology.

Keywords - **telerobotics, microsurgery, precise positioning of surgical instruments, scaling down hand motion, dexterity enhancement.**

1 Introduction

A highly skilled specialization in surgery is to perform extremely difficult procedures that require very fine controlled motions of hand-held surgical instruments. The procedures performed by these surgeons are on the eye, brain, ear, face, nerves or blood vessels and tissue features manipulated are on the order of fifty to a hundred microns in size. Micro-surgeons have a tool to help them see the instrument tips and surgical field i.e. a microscope usually used at twenty to thirty times magnification. The development of similar tool to scale down hand-to-surgical-instrument motions, and a description of its final configuration is the subject of this paper.

Development of practical telerobotics systems to assist microsurgeons [3] is a growing field of research. Although the concept is not new, advances in materials, manufacturing processes, computational power of processors and availability of miniature off-the-shelf components have enabled practical systems to become feasible. Investigators who have recently reported on micro-telerobotic workstations for bio-medical applications include Dario [6][7] Hannaford [12] and Hunter [13]. Commercial systems are also beginning to appear on the market that could potentially revolutionize the practice of micro-surgery.

A telerobotic system placed between the surgeon's hand and the surgical instrument tip allows signals between the input device and the surgical instrument to be monitored and modified. The obvious advantage obtained is the ability to scale down hand motions (in both position and orientation), thus allowing the surgeon to more precisely control the instrument tip. With a force reflecting telerobotic system, one can correspondingly amplify the forces of interaction occurring at the instrument tip to give the surgeon an enhanced sense of touch. An index switch on the master handle to activate and de-activate master-slave control allows the surgeon to keep his hand in an optimal position while covering a large workspace at the instrument tip.

There are a number of other advantages to be gained. Tremor inherent in hand motion can be identified and filtered [5],[13]. Advanced sensing and control techniques can be used to share control of the surgical instrument. For example, based on the identified location of the point of entry into an eyeball, the telerobotic system can automatically constrain the motion of the instrument to pivot about the point of entry.

An effort to demonstrate a telerobot for micro-surgery was begun at the Jet Propulsion Laboratory in October 1993 [4],[22] [23]. Our goal was to develop a telerobotic system to assist microsurgeons manipulate surgical instruments more finely than is possible with their hands. The effort was initiated in a collaboration between NASA/JPL and Micro-Dexterity Systems, Inc. The Robot-Assisted MicroSurgery (or RAMS) project concluded in October 1998 with the demonstration of 2 prototypes configured in a dual-arm bilateral force-reflecting system performing a suturing procedure. A photograph of the setup is shown on Figure 1.

The next section describes the system we developed with details of the mechanical and electronics hardware and the algorithms and advanced control software that operated the system. We conclude with a summary of our accomplishments and a description of the current status and plans.

2 System description

2.1 Overview

Figure 2 shows an overview of the hardware components of the RAMS telerobotic system.

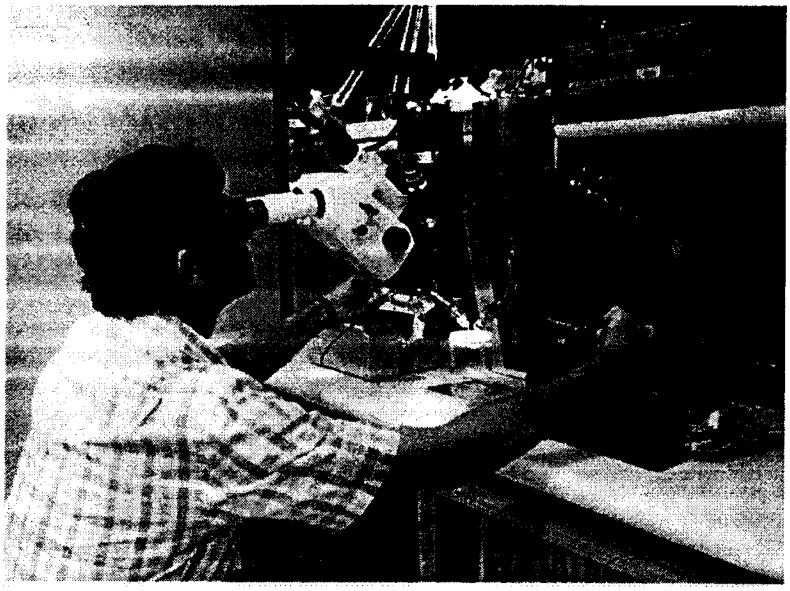


Figure 1: 2 prototypes of the RAMS system configured as a dual-arm telerobot.

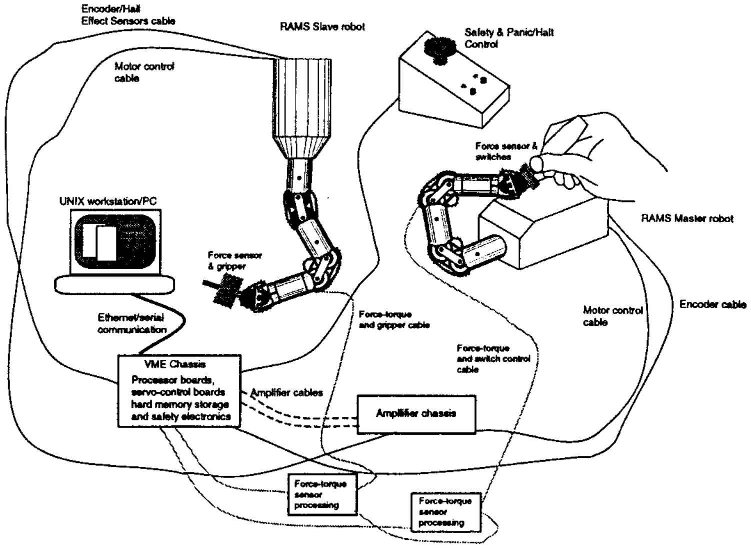

Encoder/Hall Effect Sensors cable
RAMS Slave robot
Safety & Panic/Halt Control
Motor control cable
Force sensor & switches
UNIX workstation/PC
RAMS Master robot
Force sensor & gripper
Ethernet/serial communication
Force-torque and gripper cable
Motor control cable
Encoder cable
Force-torque and switch control cable
VME Chassis
Processor boards, servo-control boards hard memory storage and safety electronics
Amplifier cables
Amplifier chassis
Force-torque sensor processing
Force-torque sensor processing

Figure 2: RAMS slave robot system.

A drawing of the interaction between the subsystems of the RAMS slave robot is shown on Figure 3.

The surgeon holds the handle of the master input device as he would a surgical instrument and commands motions for the slave-held instrument. Hand motions are read into the real-time computing system where they are processed then used to drive the slave robot. Force sensors at the instrument tip are read and processed and used to drive the master input device to give the surgeon a kinesthetic sense of interaction between the instrument tip and tissue.

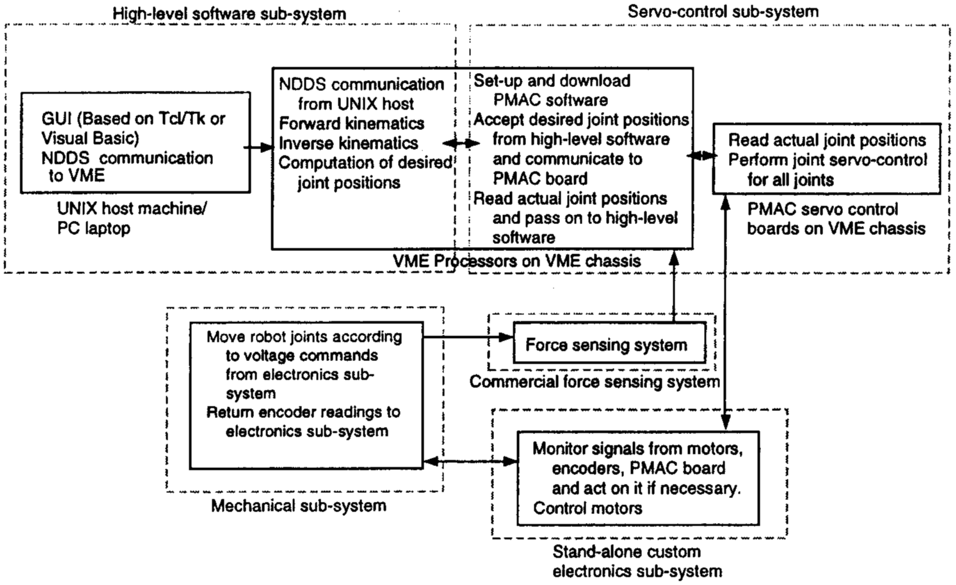

High-level software sub-system
Servo-control sub-system
GUI (Based on Tcl/Tk or Visual Basic)
NDDS communication to VME
UNIX host machine/ PC laptop
NDDS communication from UNIX host
Forward kinematics
Inverse kinematics
Computation of desired joint positions
Set-up and download PMAC software
Accept desired joint positions from high-level software and communicate to PMAC board
Read actual joint positions and pass on to high-level software
VME Processors on VME chassis
Read actual joint positions
Perform joint servo-control for all joints
PMAC servo control boards on VME chassis
Move robot joints according to voltage commands from electronics sub-system
Return encoder readings to electronics sub-system
Mechanical sub-system
Force sensing system
Commercial force sensing system
Monitor signals from motors, encoders, PMAC board and act on it if necessary.
Control motors
Stand-alone custom electronics sub-system

Figure 3: Sub-systems of the RAMS telerobot system.

A graphics user interface (GUI) implemented on a UNIX workstation or laptop PC is used to configure the parameters of the system, for example, setting ratios of position scaling or force feedback. A photograph of the RAMS system is shown on Figure 4. Details of the elements are reported in the following sections.

2.2 Mechanical

The mechanical sub-system consists of the cable-driven master input device and slave robot. The kinematics of the master and slave arms is illustrated on Figure 5. The kinematic model of the slave arm has ten joints even though it has only six degrees of freedom (dof). The kinematic model of the master has eight joints and six dof. The shoulder and elbow joints on both the master and slave robot have a unique design that constraints two joints to move as a single dof. The main advantage of this design is the independence of successive joint cable drive lengths on the position of the shoulder and elbow joints respectively. The wrist joint on the slave robot is a compact (25 millimeters in diameter) cable-drive implementation that is kinematically similar to the Ross-Heim [21] OMNI-WRIST design. The wrist kinematic model has four axes of rotation for the two independent pitch and yaw dof [26]. The advantage of this design is a large +/- 90 degrees range of motion in the slave pitch and yaw dof. The wrist on the master arm is a universal joint with two axes of rotation so it has a smaller pitch and yaw range of motion (+/- 45 degrees). Figure 5 shows the kinematics of the master and slave robots Units on the illustration are in centimeters.

All joints on the slave robot and all but the first joint (torso) of the master robot are driven by pre-tensioned antagonistic dual drives that eliminate backlash. On the slave robot,

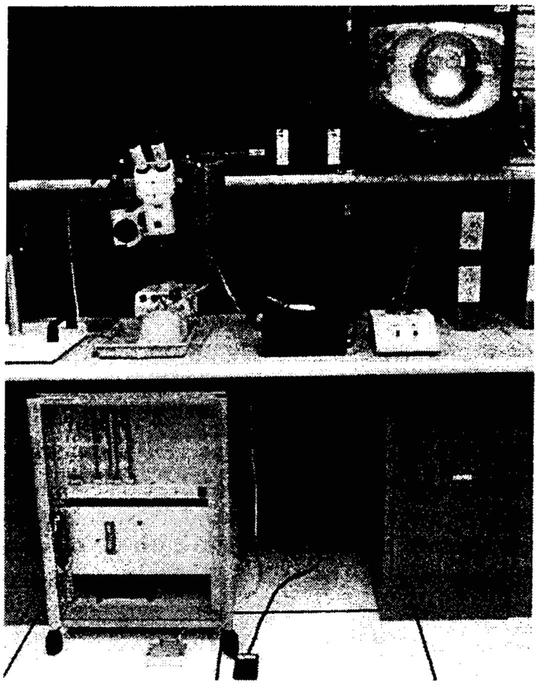


Figure 4: RAMS system.

each parallel dual drive-train consists of two or three stages of gears followed by a cable drive system. High gear ratios were selected to give the slave robot tip measurement accuracy of 1 micron per encoder count and a tip positioning accuracy of 10 microns. The requirement for ease of backdrivability of the master arm necessitated small gear ratios. Precise joint measurement and corresponding tip measurement on the master arm was achieved by using precise (10000 line) optical encoders resulting in a tip measurement accuracy of 30 microns.

The workspace of the slave robot is a 30 centimeters in diameter hemisphere while the workspace of the master is a 3 centimeters cube. Both arms are compact and lightweight. The slave arm is 2.5 centimeters in diameter and 25.8 centimeters long. Its base, which houses six brushless DC motors and six 512-line optical encoders, is 12 centimeters in diameter and 17 centimeters long. The slave robot weighs 2.5 kilogrammes. The master arm is almost identical and it is 2.5 centimeters in diameter and 24.7 centimeters long. Its base housing, which houses three brushless DC motors, three 512-line optical encoders and six 10000-line encoders, is a 23.5 centimeters by 18.4 centimeters by 10.2 centimeters box and it weighs 3.5 kilogrammes.

Mounted on the end of the slave robot is an integrated force/torque sensor and micro-gripper and drive shown on Figure 6. The force sensor is a six axes 1.7 centimeter in diameter Nano Transducer from ATI Industrial Automation. Modification of the sensor allowed mounting of a miniature gripper drive and gripper on its output and attachment to the end of the slave robot with the sensor and gripper cables flowing through the lower arm of the slave. The design minimizes the wrist to instrument tip distance for optimal instrument orientation range.

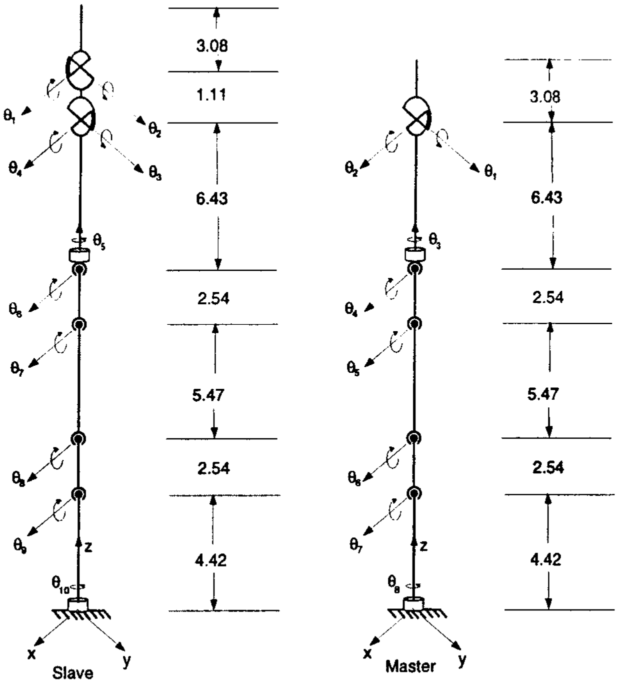


Figure 5: Slave and master robot kinematics.

A Nano Transducer is also mounted on the master arm. Attached to it is a handle shaped similar to surgical instrument handles. Micro-switches installed on the handle are used to open and close the micro-gripper on the slave robot and to activate master-to-slave control. Figure 7 shows how the handle is held.

2.3 Electronics

The RAMS electronics subsystem design includes off the shelf and custom designed electronics. Figure 8 shows a layout of its general components.

Components of the electronics subsystem are a VME chassis, an amplifier chassis, two force sensor processing chassis and safety electronics. The VME chassis houses the VME backplane and two Motorola MVME-167 computer boards used for high level system control. The VME chassis also contains the PMAC servo control cards and two supporting interface modules, power supplies (+/-15v) and a cable interface board. The VME chassis front panel contains main power control (AC) for the system. The rear panel provides access to the control computers serial communications port (RS-232). All components above are off the shelf items except the cable interface board.

The VME computer boards are the hardware portion of the high level control system. The RS-232 interface provides communication for control and observation of the robot system

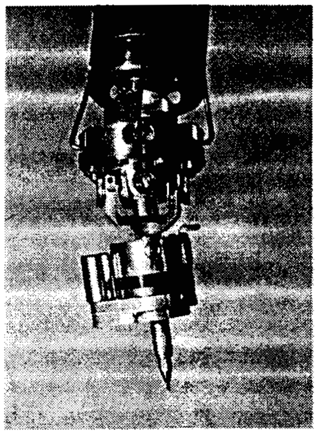


Figure 6: Slave end effector.

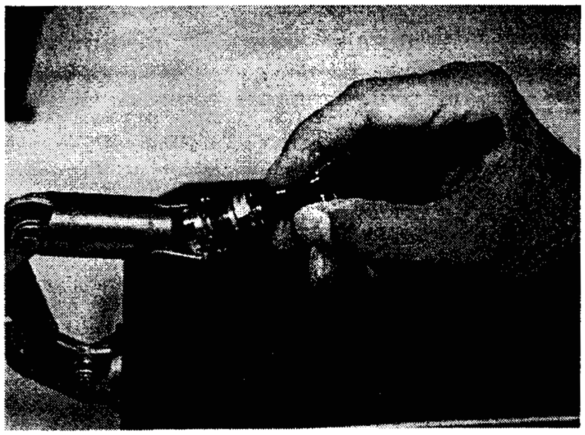


Figure 7: Master handle.

functions.

The PMAC servo boards generate 2 phase drive signals for sinusoidal commutation of the systems brushless DC motors. The linear trans-conductance amplifiers derive the third phase for the three phase motors. The PMAC receives optical encoder feedback from the motor shafts and provides low level control of the motors. The six I/O blocks and cable interface board handle signal and power distribution to the connectors on the rear panel.

The AMP (amplifier chassis) contains the six slave robot motor and three master robot motor drive amplifiers, system control electronics board, amplifier power supply and AMP subsystem power. The AMP chassis has interfaces to the VME chassis (analog inputs and control signals), the slave and master robots (motor drive signals) and to the CTRL panel subsystem (panic stop, run and initialize). The AMP chassis main power (AC) is provided by the VME chassis.

The Amplifier sub-chassis secures the individual amplifiers to the AMP chassis. This is designed to provide a thermal path to the chassis and to provide a favorable orientation with respect to the chassis air flow pattern. The system maintains the operating temperature

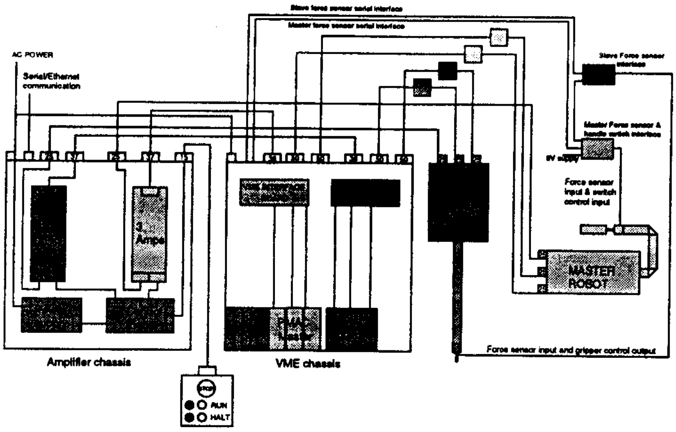

Slave force sensor serial interface
Master force sensor serial interface
AC POWER
Serial/Ethernet communication
Slave Force sensor interface
Master Force sensor & handle switch interface
5V supply
Force sensor input & switch control input
3 Amps
MASTER ROBOT
Force sensor input and gripper control output
Amplifier chassis
VME chassis
RUN
HALT

Figure 8: Electronics components and cabling.

of the amplifiers below the recommended maximum operating temperature. The frame of the Amplifier sub-chassis contains all necessary amplifier interface wiring. This makes the design highly modular to facilitate rapid check out and trouble-shooting.

The safety control electronics consists of the control electronics board and the brake relay board. The purpose of the braking function is to hold the motors in place when they are not under amplifier control. Programmable Logic Arrays (PLD) in the safety control electronics module monitors amplifier power, operator control buttons and the PANIC-HALT button, and a watchdog signal from the high-level software and control processors (indicating that they are healthy). Any anomaly triggers brakes to be set on the slave robot joint and a fault LED to be lighted. The operator must reset the safety control electronics to re-activate the system. A diagram of the safety control electronics functions and PLD state transitions are shown on Figures 9 and 10.

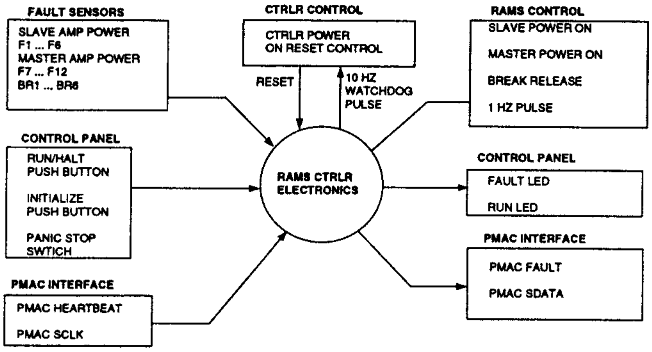
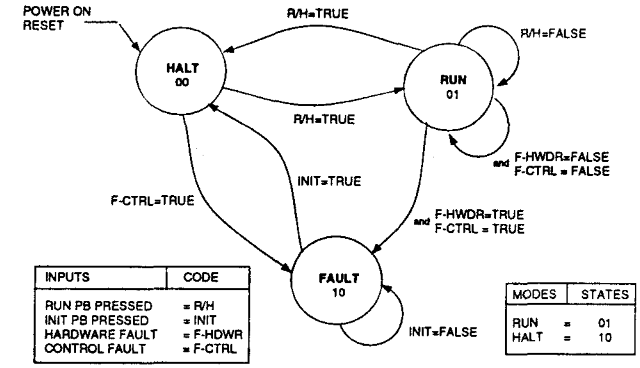


Figure 9: Function of the control electronics.



NOTE: F-CTRL=TRUE IF THERE IS A PMAC HEARTBEAT ERROR
OR IS THE PANIC SWITCH IS OPEN

Figure 10: Control electronics state transitions.

2.4 Software control

There are a number of components to the high-level software for the RAMS system and these are distributed between a graphics user interface (GUI) implemented on a UNIX workstation or an IBM PC compatible laptop computer and a VME chassis real-time system with three VME processor boards. A commercial real-time software development environment called ControlShell [16], [17] is used in the development of the software residing on the VME processors. The software is distributed between three processors to even their computational load and minimize data transfer. Software associated with the master arm and data I/O is located on one processor board, software for the slave robot on a second processor board and the servo control interface software on the third processor board. A communication package called NDDS [18] is used for low-bandwidth communication between the UNIX workstation and the real-time system and for synchronization of states of the software on the different processors. Shared memory between the processors is used for high-bandwidth data exchange between the processors. A drawing of the parts of the software is shown on Figure 11.

A Tcl/Tk [15] based GUI implemented on a UNIX workstation and communicating through an Ethernet interface was used for demonstrations and development in the laboratory environment while an equivalent Visual Basic GUI implemented on a laptop PC and communicating through a serial RS-232 interface was used for field trials. Commands from the GUI effect transitions between states in the real-time software. Transition commands have arguments that specify parameters for particular states. For example, a transition from the idle state to the master-slave control state will have a parameter that specifies the position scaling ratio to use.

The states and state transitions of the master robot are shown on Figure 12 and for the slave robot on Figure 13. The start state for both master and slave robots is their respective *idle* state. Both master and slave are initialized simultaneously by a single *initialize* command from the GUI. The master and slave *initialize* transitions return both master and slave robots back to their respective *idle* states. States of the master and slave are described on Tables 1 and 2.

At the lowest level of the real-time software are modules that perform specific functions

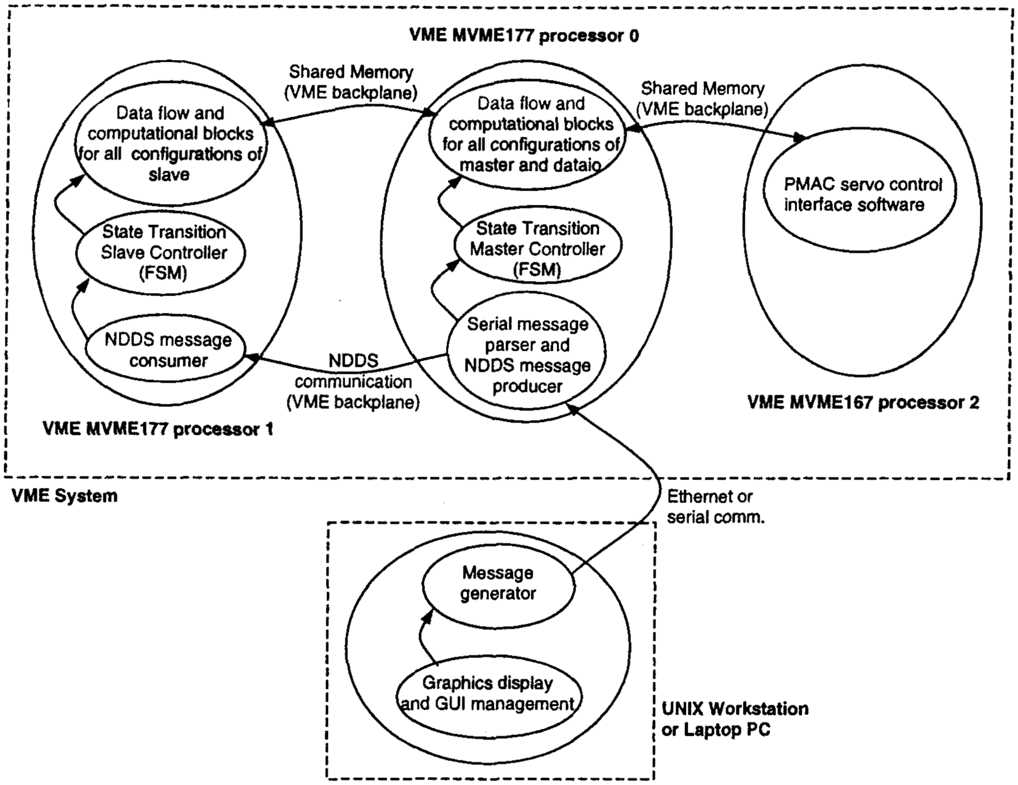


Figure 11: Parts of the high-level software.

and communicate with each other. An example of modules are the forward kinematics, inverse kinematics, filtering and coordinate transformation algorithms. There are many modules implemented and for particular states of the system, only a subset of the modules are used. Transitions activate some modules, de-activate others and re-set signals between them in the transfer from one state to another. The operation of the RAMS system corresponds to the states and state transitions of the master and slave parts of the software so we will not describe details of the software at the module level. Overview descriptions of some of the important control features of the system are described in the following section.

A background process runs on each processor board with a daisy-chain signal flow between them to finally produce a watchdog pulse output that is sent to the safety control electronics. Failure of any of the processors disrupts the watchdog signal and triggers the safety electronics to transition into its fault mode.

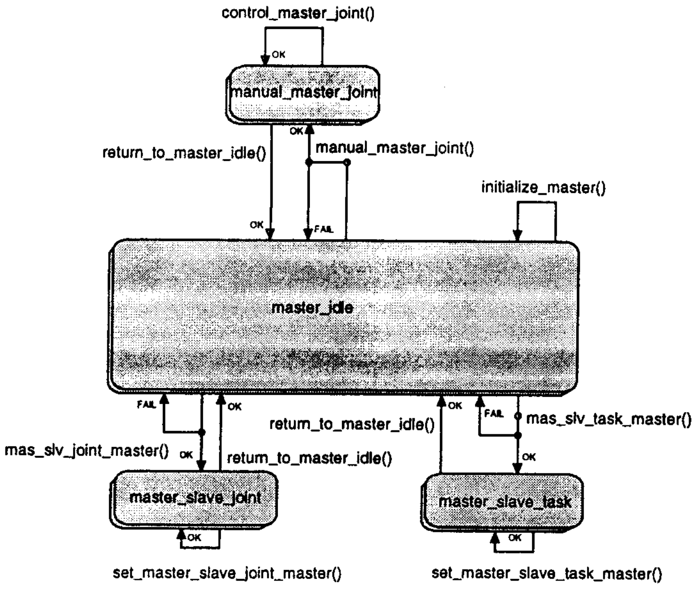


Figure 12: States and transitions of the RAMS master real-time software.

2.5 Control feature highlights

2.5.1 Force feedback

The primary mode of operation of the RAMS system is in its master_slave_task mode where the master handle motions control the slave instrument tip and forces of interaction at the instrument tip are fed back to the master handle. A system having this behavior has been referred to as a bilateral force reflecting telemanipulator and it has been the subject of research since the 1950s [2] [9] [11].

We use a fairly simple and conventional control method to achieve this behavior. A diagram of the signal flow in the master-slave control mode with force feedback is shown on Figure 14 and a summarized description is as follows. The master robot forward kinematics algorithm computes the master handle position and orientation from master joint positions read in from the servo control cards through shared memory. The handle incremental motion is computed, filtered and passed to the slave processor. The incremental handle position is scaled (based on a GUI parameter setting) and transformed into one of a couple of coordinate frames (depending on a parameter set on the GUI), then modified if necessary by a pivoting algorithm to be described in the next section. The incremental motion is added to the slave tip position to produce the desired slave tip position. If the active switch on the master handle is on indicating that the operator desires to control the slave arm, the signal flows into an inverse kinematics algorithm (implemented using a technique from [19][20]) that converts the desired motions of the slave tip into its equivalent slave joint motions. These joint motions are used to command the joints of the slave through shared memory and the

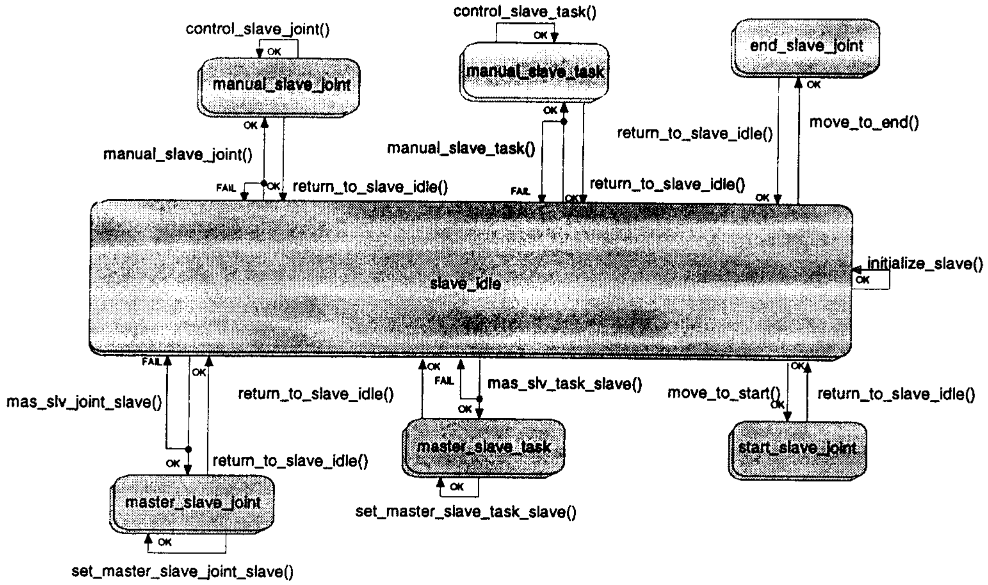


Figure 13: States and transitions of the RAMS slave real-time software.

servo control boards.

Force and torque sensor readings on the slave robot are transmitted through shared memory to the master processor. The signals are transformed to the master arm coordinate frame and filtered to reduce noise in the signal. The force signal is then multiplied by the transpose of the master robot Jacobian matrix to produce the equivalent joint torques. The joint torque values are multiplied by a set of gains to compute the equivalent D/A value that would produce the desired torque on the joint. These values are sent through shared memory to the joint torque controller on the servo control boards.

2.5.2 Shared pivoting control

One of the demonstrations performed with the RAMS system was to use it to remove a microscopic particle from within a simulated eyeball. This required the insertion of a pair of micro-forceps mounted on the slave robot into the eyeball and control of its motion to approach and grasp the particle and then withdraw without causing any damage to the eyeball. A shared control mode of master-slave operation of the RAMS system was implemented to assist the user when performing the demonstration (see Figure 15). This feature was necessary because at the time of the demonstration, instrument force feedback was not implemented so the user could not use guiding forces from the pivot point on the instrument shaft to assist in the procedure.

The goal of the shared control algorithm is for the software to autonomously control the pitch and yaw motion of the instrument upon entry into the simulated eyeball. It leaves

State	Description
master_idle	Master arm joints are uncontrolled and free to be moved. Forward kinematics is computed continuously to determine master handle position.
manual_master_joint	A debugging mode of control of the master arm in which the user can set desired joint positions on the GUI and the servo control system controls the master joints to the specified position. The slave robot remains in its *slave_idle* state when the master arm is in this mode.
master_slave_joint	A mode of control in which incremental changes in the joint positions of the master are used as incremental changes commanded to the corresponding slave joints. This mode was used in debugging during development. The slave real-time software is also in its corresponding *master_slave_joint* state when this state is active on the master.
master_slave_task	A mode in which the incremental motions of the master handle position and orientation are used to command corresponding slave tip incremental motions. This is the mode in which the system is used for demonstration of telerobotic operation. The slave real-time software is also in its corresponding *master_slave_task* state when this state is active on the master.

Table 1: Description of the states of the master real-time software.

control of the translation motions of the instrument tip and rotation motion about the instrument shaft to the operator. It also limits the depth of entry of the instrument to prevent puncture of the inside wall of the eyeball.

Many alternatives are possible to solve the constraint robot control problem. Related work include generalized optimization approaches to the solution of the motion control of redundant robots proposed by many researchers [1], [14], [25]. A method of imposing absolute constraints for manipulators has also been proposed [8]. We chose to develop a alternative and specific solution for our application that is simple and requires minimal computation.

The procedure in using the pivoting algorithm is to move towards the entry port perpendicular to the eyeball surface until the tip of the forceps touches the eyeball at the entry point. A button on the GUI is selected to indicate the position of the forceps with respect to the eyeball and to turn on the shared control pivoting algorithm. The operator then continues to insert the forceps into the eyeball and position its tip to approach the microscopic particle and grasp it. The control software adjusts the forceps pitch and yaw to always have it pivot about the entry point. The operator opens the forceps and grasps the particle and then withdraws the forceps while in this shared control mode. The shared control mode is

State	Description
slave_idle	Slave arm joints are controlled to the positions they have at the entry into this *slave_idle* state.
start_slave_joint	This state is an autonomous move through a pre-defined trajectory from the start position to a desired working position of the slave robot. A transition from this state to the *slave_idle* state is automatically issued when the final position is reached. The master robot remains in its *master_idle* state when the slave arm is in this mode.
end_slave_joint	This state is an autonomous move through a pre-defined trajectory from the working position to a defined home position of the slave robot. A transition from this state to the *slave_idle* state is automatically issued when the home position is reached. The master robot remains in its *master_idle* state when the slave arm is in this mode.
manual_slave_joint	A debugging mode of control of the slave arm in which the user can set desired joint positions on the GUI and the servo control system controls the slave joints to the specified position. The master robot remains in its *master_idle* state when the slave arm is in this mode.
manual_slave_task	A debugging mode of control of the slave arm in which the user can set desired tip positions on the GUI and the servo control system controls the slave joints so that the slave tip moves to the desired position. The master robot remains in its *master_idle* state when the slave arm is in this mode.
master_slave_joint	A mode of control in which incremental changes in the joint positions of the master are used as incremental changes commanded to the corresponding slave joints. This mode was used in debugging during development. The master real-time software is also in its corresponding *master_slave_joint* state when this state is active on the slave.
master_slave_task	A mode in which the incremental motions of the master handle position and orientation are used to command corresponding slave tip incremental motions. This is the mode in which the system is used for demonstration of telerobotic operation. The master real-time software is also in its corresponding *master_slave_task* state when this state is active on the slave.

Table 2: Description of the states of the slave real-time software.

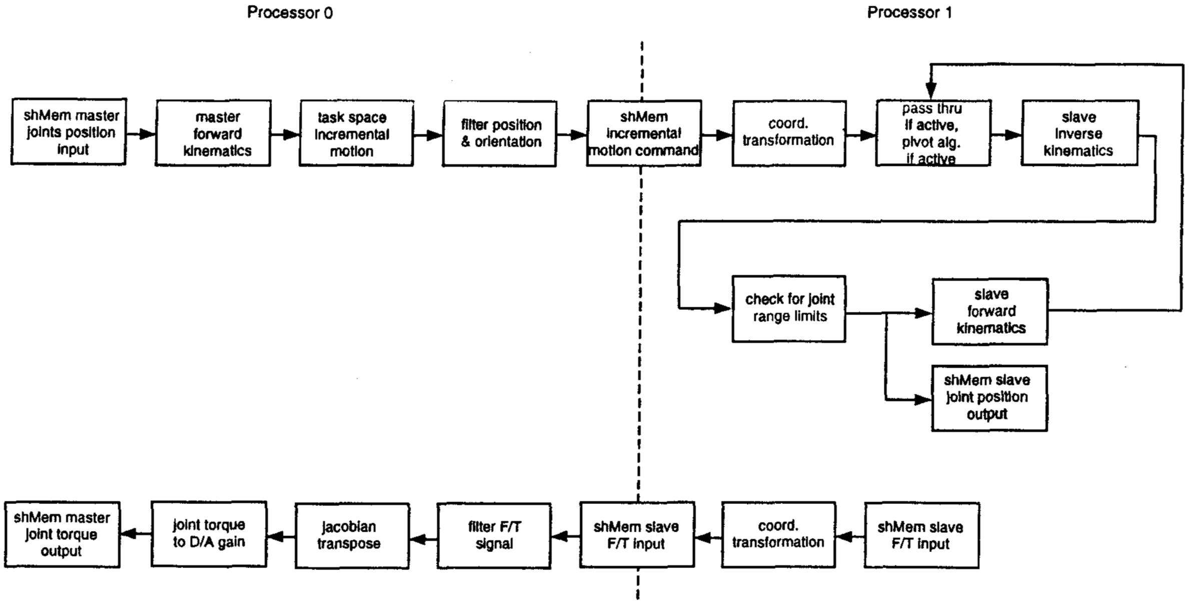

Processor 0
Processor 1
shMem master joints position input
master forward kinematics
task space incremental motion
filter position & orientation
shMem incremental motion command
coord. transformation
pass thru if active, pivot alg. if active
slave inverse kinematics
check for joint range limits
slave forward kinematics
shMem slave joint position output
shMem master joint torque output
joint torque to D/A gain
Jacobian transpose
filter F/T signal
shMem slave F/T input
coord. transformation
shMem slave F/T input

Figure 14: Signal flow for master-slave control with force feedback.

terminated by selecting the button on the GUI again when the tip of the forceps is at the entry point on its way out of the eyeball. The operator can then freely control the forceps in 6 dof.

Two acceptable assumptions made in the pivoting algorithm are:

- the eyeball is spherical in shape with a known diameter [10].
- the user positions the forceps perpendicular to the eyeball surface at the entry point at the initiation of the pivoting procedure.

The algorithm used is described with the illustration shown on Figure 16. The change in the pitch (rotation about x axis) and yaw (rotation about y axis) angles are given by

$$\Delta\theta_x = -\Delta y/l \quad (1)$$

$$\Delta\theta_y = \Delta x/l \quad (2)$$

where $\Delta\theta_x$ is the change in pitch angle, $\Delta\theta_y$ is the change in yaw angle, Δx is the commanded change in the x-axis coordinate of the instrument tip, Δy is the commanded change in the y-axis coordinate of the instrument tip and l is the length from the instrument tip to the pivot point. Two conditions are imposed on the input move command Δx, Δy and Δz prior to the computation of the changes to the pitch and yaw angles above. The first condition is to ensure that the commanded forceps tip position is within the *spherical* boundaries of the eyeball. If the commanded position is outside the boundary, the forceps move command is ignored. At locations of the forceps tip within the eyeball but close to the entry point,

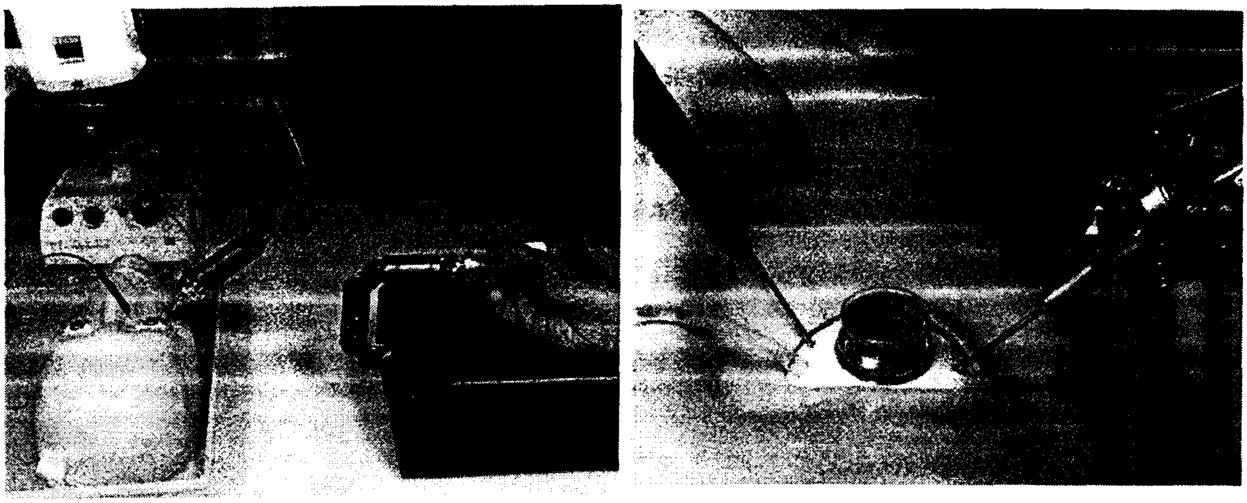


Figure 15: Two views of the eye microsurgery demonstration with the RAMS system.

equations (1) and (2) above become singular due to the small values of the pivot length l. Therefore, a second modification is performed to increasingly constrain translational motion commands in the pivot point referenced coordinate frame X-Y plane when the forceps tip is close to the entry point and apply equations (1) and (2) above only within that constraint. The physical constraint is a horn-shaped boundary as illustrated on Figure 16. The algorithm thus maintains the instrument shaft to be normal to the eyeball surface when its tip is close to the entry point but allows increasingly greater freedom to pivot as the instrument probes deeper into the eyeball. This prevents large pitch or yaw angles from being computed due to small x or y translations when close to the entry point. The resulting behavior worked very well with our engineers as operators. It was used successfully in a number of demonstrations of the removal of a particle from the simulated eyeball. One author (Dr. Charles, an vitreo-retinal eye surgeon), however, found the shared pivoting control to be unnatural because it eliminated the ability to control the position of the pivot point to orient the eyeball for a view through the pupil at the peripheral retina.

3 Conclusion

Results from performance tests of the RAMS system conducted at the USC School of Medicine found significant improvement in precise positioning of instruments in well trained subjects at the cost of increased time needed to perform the positioning task. In addition, field tests on early prototypes of the RAMS system were also conducted at the Cleveland Clinic Foundation [24] and at the Manhattan Veterans Administration Medical Center.

Two prototype RAMS systems have been built at JPL and they have been used in a dual

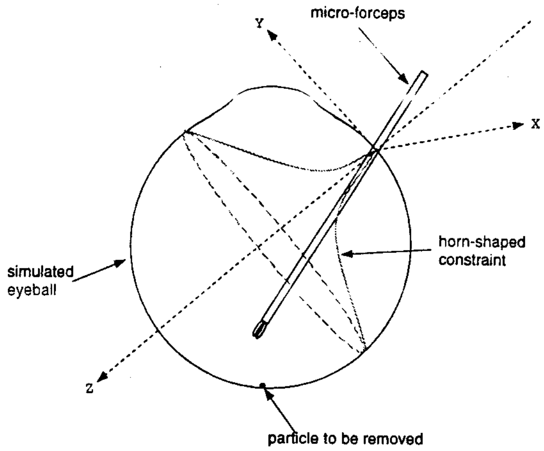


Figure 16: Pivoting shared control.

arm configuration to demonstrate a micro-suturing procedure (see Figures 1 and 17).

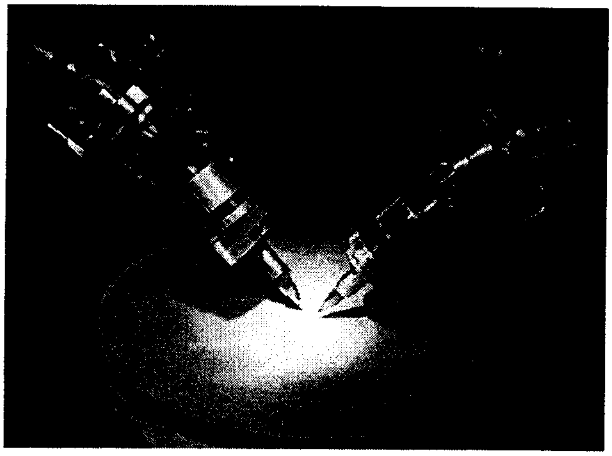


Figure 17: Close-up view of dual-arm telerobotic micro-suturing.

The RAMS effort has shown the potential for extending the capabilities of microsurgeons and for enabling new procedures that are beyond the skill of the best surgeons. We have demonstrated the practical utility of the RAMS system in simulated and realistic microsurgical settings. MicroDexterity Systems continues its development of this technology to seek FDA certification leading to commercial products.

References

[1] J. Baillieul, "Kinematic programming alternatives for redundant manipulators," in Proceedings of the IEEE International Conference on Robotics and Automation, pp 722-728, St. Louis, MO, 1985.

[2] A. K. Bejczy, J. K. Salisbury, "Kinesthetic coupling between operator and remote manipulator," in Proceedings of the 1980 International Conference on Robotics and Automation, pp 1073-1080, 1980.

[3] S. Charles, "Dexterity Enhancement for Surgery," in Computer Integrated Surgery: Technology and Clinical Applications, ed. R. H. Taylor, S. Lavalle, G. Burdea, R. Mosges, MIT Press, Cambridge, MA 1996.

[4] S. Charles, H. Das, T. Ohm, C. Boswell, G. Rodriguez, R. Steele, and D. Istrate "Dexterity-enhanced Telerobotic Microsurgery," Presented at NASA University Centers Conference, February, 1997, Albuquerque, NM and 8th International Conference on Advanced Robotics (ICAR '97), July 1997, Monterey, CA.

[5] S. Charles, personal communication

[6] P. Dario, M. C. Carroza, L. Leniconi, B. Magnani, S. D'Attanasio "A Micro Robotic System for Colonoscopy," in Proceedings of the 1997 International Conference on Robotics and Automation, Albuquerque, New Mexico, April, 1997.

[7] P. Dario, C. Pagetti, N. Troisfontaine, E. Papa, T. Ciucci, M. C. Carrozza, M. Maracci "A Miniature Steerable End-effector for Application in an Integrated System for Computer-assisted Arthroscopy," in Proceedings of the 1997 International Conference on Robotics and Automation, Albuquerque, New Mexico, April, 1997.

[8] J. Funda, R. Taylor, B. Eldridge, S. Gomory, and K. Gruben, "Constrained Cartesian motion control for teleoperated surgical robots," IEEE Trans. on Robotics and Automation v12, n3, pp 453-465, June 1996.

[9] R. Goertz, F. Bevilacqua, "A force-reflecting positional servomechanism," Nucleonics, V10, n11, pp 43-45, 1952.

[10] H. Gray, L. H. Bannister, M. M. Berry, P. L. Williams, "Gray's Anatomy," 38th Edition, Pub: Churchill Livingstone, October 1995.

[11] M. Handlykken, T. Turner, "Control system analysis and synthesis for six-degree-of-freedom universal force reflecting hand controller," in Proceedings of the IEEE Conference on Decision and Control, 1980.

[12] B. Hannaford, J. Hewitt, T. Maneewarn, S. Venema, M. Appleby, R. Ehresman "Telerobotic Remote Handling of Protein Crystals," in Proceedings of the 1997 International Conference on Robotics and Automation, Albuquerque, New Mexico, April, 1997.

[13] I. W. Hunter, T.D. Doukoglou, S.R. Lafontaine, P.G. Charette, L.A. Jones, M.A. Sagar, G. D. Mallinson, P.J. Hunter "A Teleoperated Microsurgical Robot and Associated Virtual Environment for Eye Surgery," Presence, v 2, n 4, pp 265-280, Fall 1993.

[14] C. A. Klein, Huang, C-H., "Review of Pseudo-inverse Control for use with Kinematically Redundant Manipulators," IEEE Trans. on Systems, Man and Cybernetics, v. SMC-13, n. 3, March, 1983.

[15] J. K. Ousterhout, "Tcl and the Tk Toolkit," Addison Wesley, Reading, Mass. 1994.

[16] Real-time Innovations, Inc., "Control Shell Programmer's Reference Manual Vol. 1," Sunnyvale, CA, 1995.

[17] Real-time Innovations, Inc., "Control Shell Programmer's Reference Manual Vol. 2," Sunnyvale, CA, 1995.

[18] Real-time Innovations, Inc., "NDDS Programmer's Reference Manual," Sunnyvale, CA, 1995.

[19] G. Rodriguez, K. Kreutz, and A. Jain, "A Spatial Operator Algebra for Multibody System Dynamics," The Journal of the Astronautical Sciences, Vol. 40, No. 1, pp. 27-50, January-March 1992.

[20] G. Rodriguez, "Kalman Filtering, Smoothing, and Recursive Robot Arm Forward and Inverse Dynamics," IEEE Transactions on Robotics and Automation, Vol. 3, pp. 624-639, Dec. 1987.

[21] M. E. Rosheim, "Robot Wrist Actuators," John Wiley & Sons., New York, 1989.

[22] P. S. Schenker, Das, H., and Ohm, T. "A new robot for high dexterity microsurgery" Proceedings of the First International Conference, CVRMed '95, Nice, France April, 1995. also in Computer Vision, Virtual Reality and Robotics in Medicine, Lecture Notes in Computer Science, Ed. Nicholas Ayache, Springer-Verlag, Berlin 1995.

[23] P. S. Schenker, S. Charles, H. Das and T. Ohm "Development of a Telemanipular for Dexterity Enhanced Microsurgery," Proceedings of the Second Annual International Symposium on Medical Robotics and Computer Assisted Microsurgery, November, 1995, Baltimore, MD.

[24] M. Siemionow, "Robot Assisted MicroSurgery: experience with the RAMS system in plastic surgery research," Minimally Invasive Coronary Artery Bypass Grafting Workshop, Sept. 1997, Utrecht, The Netherlands.

[25] D. E. Whitney, "Resolved Motion Rate Control of Manipulators and Human Prostheses,", IEEE Trans on Man-Machine Systems, v. MMS-10, n. 2, June, 1969.

[26] R. L. Williams III, "Forward and Inverse Kinematics of Double Universal Joint Robot Wrists," Space Operations, Applications and Research (SOAR) Symposium, Albuquerque, NM, June 26-28, 1990.

Acknowledgment

This work was carried out at the Jet Propulsion Laboratory under contract with the National Aeronautics and Space Administration. The authors affiliated with JPL are in the Automation and Control Section, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109. Steve Charles, MD is the CEO of MicroDexterity Systems, Inc.